# **Slide 2**

Et pour présenter notre travail on va passer par 5 points, premièrement une présentation générale de la solution, et ensuite on passe par la phase de la conception, et ensuite la partie de développement, et après qu’est-ce que on a fait pour sécuriser la solution, et le dernier point sera la conclusion.

# **Slide 3**

Tout d’abord la présentation générale.

# **Slide 4**

Notre travail consiste à développer un site web pour l’ISCAE, en utilisant le plateforme .NET, Ce site permet l’interaction entre les étudiants, les professeurs et l’administration, en respectant les recommandations de la sécurité d’une application web, et en utilisant le responsive web design pour s’adapter à la gamme d’appareil de l’utilisateur, et que cette solution optimise les ressources physiques et finalement assurer la maintenabilité de la solution.

# **Slide 7**

Un outil de développement est un logiciel qui aide un développeur dans le déroulement d’une activité de développement, et les outils que nous avons utilisés sont :

JMerise : dédié à la modélisation des MCD pour Merise.

Microsoft SQL Server 2016 : un SGBDR qui offre plusieurs avantages dont la scalabilité, la redondance, la génération de rapports et la synchronisation.

SQL Server Management Studio 2016 : un environnement intégré qui permet d'avoir accès, de configurer, de gérer, d'administrer et de développer tous les composants de SQL Server.

Visual Studio 2015 : un véritable « couteau suisse » dans le monde du développement, il permet de développer tout type d’application : Cloud, mobile, web, multiplateforme, desktop …etc.

GIT : C’est un logiciel de gestion de versions très puissant et récent, créé par Linus Torvalds le créateur du système d’exploitation LINUX.

Bitbucket : c’est une service web d'hébergement et de gestion de développement logiciel utilisant les logiciels de gestion de versions Git et Mercurial.

# **Slide 8**

Le but de ces principes est d’améliorer le design de nos applications et ainsi les rendre plus facilement maintenable et lisible et faciliter les éventuelles évolutions.

SOLID est un acronyme introduit au début des années 2000 par Robert Cecil Martin. Qui est dans l’industrie des logiciels depuis 1970 c’est aussi l’inventeur de la méthode agile pour la gestion des projets et le créateur de l’organisation Clean Coders.

Bon Solid résume 5 principes clés à appliquer :

**Single Responsability Principle :** une classe ne doit posséder qu’une et une seule responsabilité.

**Open Closed Principle :** Les entités logicielles doivent être ouvertes aux extensions, mais fermées aux modifications.

**Liskov Substitution Principle :** assurer que notre hiérarchie suit un fonctionnement logique et cohérent.

**Interface Segregation Principle :** utiliser les interfaces pour définir des contrats, des ensembles de fonctionnalités répondant à un besoin fonctionnel.

**Dependency** **Inversion Principle :** Les modules de haut niveau ne doivent pas dépendre des modules de bas niveau. Les deux doivent dépendre d'abstractions.

# **Slide 9**

La séparation par couches de responsabilités sert à découpler au maximum une couche de l'autre afin d'éviter l'impact d'évolutions futures et la maintenabilité de la solution.

À cet effet on a découpé la solution en 3 couches :

Couche DAL : chargé de l’accès aux données persistantes.

Couche BLL : contient les traitements représentant les règles métier.

Couche présentation : contient les différents types de clients, léger ou lourd.

**Slide 13**

Dans ce couche nous allons présenter 4 points :

ASP.NET

Le Pattern MVC

Web API

Et finalement Ninject

Premièrement ASP.NET est la plateforme de développement de Microsoft permettant la réalisation des applications web.

ASP.NET propose deux logiques pour construire une application web soit WebForms ou MVC.

# **Slide 14**

Nous avons choisi ASP.NET MVC parce que elle offre un cadre de réalisation des applications web, grâce à ce patron de conception réputé, il donne également plus de liberté dans le rendu du HTML.

L’utilisateur envoi une requête qui demande l’affichage d’une action et grâce à la fonctionnalité nommé le routage ASP.NET qui est utilisé par le framework MVC pour router les requêtes vers les actions des contrôleurs, et ce dernier va appeler les modèles correspondant et il va créer une vue et il l’envoi à l’utilisateur, on peut aussi appliquer des filtres avant ou après l’exécution d’une action.

# **Slide 18**

Le projet OWASP Top Ten : Le but de ce projet est de fournir une liste des dix risques de sécurité applicatifs Web les plus critiques. La majorité des audits de sécurité informatique Web sont basées sur ce Top Ten.

**Oubli de valider les entrées des utilisateurs :** pour prévenir cette faille on a utilisé **La méthode ValidateInput**

**Contrôle d'accès inefficace :** pour prévenir cette faille on a utilisé **ActionFilter, la vérification des données dans** **le couche BLL**

**Mauvaise gestion des sessions :** pour prévenir cette faille on a utilisé **un gestionnaire de session**

**Cross Site Scripting (XSS) :** pour prévenir cette faille on a utilisé la bibliothèque **AntiXSS**

**Dépassement de mémoire tampon :** .net ne subit pas de cette vulnérabilité

**Injection de commandes :** L’utilisation du MVC, le découpage en 3 couches et l’utilisation du principe O/RM toutes ces procédures aident à prévenir cette vulnérabilité.

**Mauvaise gestion des erreurs :** pour prévenir cette faille on a utilisé **un gestionnaire d'erreurs globales au niveau de l'application**

**Mauvaise utilisation du chiffrement :** on a utilisé seulement **le hachage** des mots de passes des utilisateurs

**Failles dans l'administration distante :** pour prévenir cette faille on a utilisé **ActionFilter**

**Mauvaise configuration du serveur web et des applications :** pour prévenir cette faille on a amélioré la sécurité du fichier **Web.config**, et on a proposé les bonnes pratiques pour sécuriser le **serveur web IIS**.

# **Slide 19**

Le projet peut être amélioré, et ce, en lui ajoutant la bibliothèque ***SignalR*** de Microsoft pour gérer les notifications en temps réel, en lui ajoutant aussi les contrôleurs ***Web API*** et le développement d’une application mobile, et encore la solution est ***extensible*** c'est-à-dire qu'il pourra y avoir une possibilité d'ajouter ou de modifier de nouvelles fonctionnalités.

Finalement nous adressons nos vifs remerciements à notre encadreur Dr. Ethmane Ould Ebbe pour son encadrement, son soutien sans failles et sa disponibilité, nous tenons également à remercier et exprimer notre profond respect aux membres de jury d’avoir accepté de juger ce travail.

MERCI